



ARL-TR-8246 • DEC 2017



# Using a Regression Method for Estimating Performance in a Rapid Serial Visual Presentation Target-Detection Task

by Jonroy D Canady, Benjamin T Files, and Amar R Marathe

## **NOTICES**

### **Disclaimers**

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



# **Using a Regression Method for Estimating Performance in a Rapid Serial Visual Presentation Target-Detection Task**

**by Jonroy D Canady, Benjamin T Files, and Amar R Marathe**  
*Human Research and Engineering Directorate, ARL*

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YYYY) December 2017		2. REPORT TYPE Technical Report		3. DATES COVERED (From - To) May 2016–October 2017	
4. TITLE AND SUBTITLE Using a Regression Method for Estimating Performance in a Rapid Serial Visual Presentation Target-Detection Task				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Jonroy D Canady, Benjamin T Files, and Amar R Marathe				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory Human Research and Engineering Directorate (ATTN: RDRL-HRF-A) Aberdeen Proving Ground, MD 21005-5068				8. PERFORMING ORGANIZATION REPORT NUMBER  ARL-TR-8246	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>Estimating target-detection performance in the rapid serial visual presentation (RSVP) target-detection paradigm can be challenging when the interstimulus interval is small relative to the variability in human response time. The challenge arises because assigning a particular response to the correct image cannot be done with certainty. Existing solutions to this challenge establish a heuristic for assigning responses to images and thereby determining which responses are hits and which are false alarms. We developed a regression-based method for estimating hit rate and false-alarm rate that corrects for expected errors in a likelihood-based assignment of responses to stimuli. Simulations show that this regression method results in an unbiased and accurate estimate of target detection performance. The regression method had lower estimation error than 3 existing methods, and, in contrast to the existing methods, the errors made by the regression method do not depend strongly on the true values of hit rate and false-alarm rate. Based on its better estimation of hit rate and false-alarm rate, the regression method proposed here appears to be the best choice when estimating the hit rate and false-alarm rate is the primary interest.</p>					
15. SUBJECT TERMS RSVP, hit rate, false-alarm rate, response time, simulation					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT  UU	18. NUMBER OF PAGES  48	19a. NAME OF RESPONSIBLE PERSON Jonroy D Canady
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) (410) 278-0720

## Contents

---

<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vi</b>
<b>1. Summary</b>	<b>1</b>
<b>2. Introduction</b>	<b>1</b>
<b>3. Methods, Assumptions, and Procedures</b>	<b>3</b>
3.1 Estimation Methods	3
3.1.1 Established Methods for Estimating HR and FAR	3
3.1.2 Proposed Method	4
3.1.3 Using the Proposed Method	5
3.2 Evaluation Methods	6
3.2.1 Participants	6
3.2.2 Stimuli and Procedure	7
3.2.3 Simulations	7
<b>4. Results and Discussion</b>	<b>10</b>
4.1 Illustrative Subset of Results	10
4.2 Full Results	11
4.3 Statistical Assessment of Bias in Estimation	13
4.4 Simulations Run with Inaccurate RT-PDF Estimates	16
4.5 Analyzing Experiment Data	18
4.6 Discussion	19
<b>5. Conclusions</b>	<b>22</b>
<b>6. References</b>	<b>24</b>
<b>Appendix. Rapid Serial Visual Presentation Performance (RSVP) Estimator (RPE) Package Source Code</b>	<b>27</b>

<b>List of Symbols, Abbreviations, and Acronyms</b>	<b>39</b>
<b>Distribution List</b>	<b>40</b>

## List of Figures

- Fig. 1 Timelines illustrating existing response assignment methods. Blue hash marks indicate onset times of nontarget stimuli; red hash marks indicate onset times of target stimuli. Green hashes indicate times at which a response occurred. Interstimulus interval is 0.5 s. A) In the window method, a window of time (typically 0–1 s posttarget) is established. Responses falling within that window are declared hits. In this example, the first and third responses would be classified as hits, and the second would be classified as a false alarm. B) The same experiment timeline as analyzed with the distribution method. The black curves are the response time probability density function reversed and with its origin at the times of response. Numbers below stimulus hashes show the attribution resulting from the corresponding response, as computed using Eq. 1. Using maximum likelihood (the max method) assigns the response to the stimulus with the highest likelihood. .... 3
- Fig. 2 Simulation method; the process for one iteration of the simulation. It was repeated 250 times per combination of HR and FAR. Analysis was done separately using each of the 4 analysis methods described in the text..... 8
- Fig. 3 Estimation method performance examples. These plots illustrate estimation results for specific combinations of simulated HR and FAR when the true probability mass function of the response times was known. The pairs HR 0.50, FAR 0.10; HR 0.80, FAR 0.02; and HR 0.99, FAR 0.01 were selected as illustrative of poor, good, and excellent RSVP target-detection performance, respectively. Bars show the median estimation error, and error bars show plus/minus one standard deviation for each of the 4 estimation methods at 3 presentation rates. The upper row of panels show HR estimation errors, and the lower row shows FAR estimation errors. .... 11
- Fig. 4 HR estimation error summary. Each panel shows the simulation results for one of the estimation methods (columns) at a particular presentation rate (rows) when the true PDF of the response times was known. Colors indicate the difference between the median estimate of the HR and the simulated value of HR, clipped to an absolute value of 0.2. Within a panel, simulated FAR increases from left to right, and simulated HR increases from bottom to top. All methods except the regression method have HR estimation errors that clearly depend on the simulated HR and FAR, and the overall magnitude of errors increases as the presentation rate increases. .... 12
- Fig. 5 FAR estimation error summary. Each panel shows the simulation results for one of the estimation methods (columns) at a particular presentation rate (rows) when the true PDF of the response times was known. Colors indicate the difference between the median estimate of the FAR and the simulated value of FAR clipped to an absolute value

	of 0.1. Within a panel, simulated FAR increases from left to right, and simulated HR increases from bottom to top. All methods except the regression method have FAR estimation errors that clearly depend on the simulated HR and FAR, and the overall magnitude of errors increases as the presentation rate increases. ....	12
Fig. 6	Estimation method performance examples with flat RT-PDF .....	16
Fig. 7	HR estimation error summary with flat RT-PDF .....	17
Fig. 8	FAR estimation error summary with flat RT-PDF .....	17
Fig. 9	Estimated HRs from experimental data with 4 different estimation methods. HR was estimated from the response data from 15 subjects using the distribution (d), max (x), window (w), and regression (r) methods. The colors for individual subject data are based on the estimates from the regression method to illustrate how the relative ordering of subjects changes based on estimation method. ....	19

## List of Tables

Table 1	HR estimate performance with accurate RT-PDF .....	13
Table 2	FAR estimate performance with accurate RT-PDF .....	13
Table 3	Omnibus ANOVA for HR estimation with accurate RT-PDF .....	14
Table 4	Method-specific ANOVA window method .....	15
Table 5	Method-specific ANOVA max attribution method .....	15
Table 6	Method-specific ANOVA distribution method .....	15
Table 7	Method-specific ANOVA regression method .....	16
Table 8	HR estimate performance with flat RT-PDF .....	18
Table 9	FAR estimate performance with flat RT-PDF .....	18

## 1. Summary

---

Estimating target-detection performance in the rapid serial visual presentation (RSVP) target-detection paradigm can be challenging when the interstimulus interval is small relative to the variability in human response time. The challenge arises because assigning a particular response to the correct image cannot be done with certainty. Existing solutions to this challenge establish a heuristic for assigning responses to images and thereby determining which responses are hits and which are false alarms.

We developed a regression-based method for estimating hit rate and false-alarm rate that corrects for expected errors in a likelihood-based assignment of responses to stimuli. Simulations show that this regression method results in an unbiased and accurate estimate of target-detection performance. The regression method had lower estimation error than 3 existing methods, and in contrast to the existing methods, the errors made by the regression method do not depend strongly on the true values of hit rate and false-alarm rate. The most commonly used existing method performed well when simulated performance was nearly perfect, but not when behavioral error rates increased.

Based on its better estimation of hit rate and false-alarm rate, the regression method proposed here would seem the best choice when estimating the hit rate and false-alarm rate is the primary interest.

## 2. Introduction

---

Finding target images in large databases of candidate images is a difficult problem, and while computer vision algorithms are adequate for some tasks, for others human vision is required. A key insight to approaching this problem is that humans tasked with finding target images achieve high target-detection accuracy even if the images are shown very rapidly (Intraub 1981). Using RSVP with images displayed at rates of 2–10 Hz can dramatically increase the rate at which target images are found in image databases compared with self-paced image viewing (Mathan et al. 2006; Parra et al. 2008). This is a somewhat different use of RSVP from its classical use as a tool for investigating the time course of perception (Potter and Levy 1969; Chun and Potter 1995; Keyser et al. 2001; Nasanen et al. 2006), with particular focus on the attentional blink (Raymond et al. 1992) and repetition blindness (Kanwisher 1987) phenomena. In those uses of RSVP, a typically short stream of words or images is displayed and then the viewer is asked one or more questions about the just-viewed RSVP stream. Here, instead, images are presented

continuously, and the viewer is asked to press a button immediately in response to images containing a target of interest.

In practical applications of the RSVP target-detection paradigm, the goal is to identify images that are targets in a potentially large database of unknown images. However, in experiment settings, the identity of images are known, and the question is how well a human subject performs the target-detection task. RSVP target-detection task performance can be difficult to quantify due to response time variability (Mathan et al. 2006; Sajda et al. 2010). This report introduces a novel method for estimating performance on the RSVP target-detection task in experimental settings in which image labels are known.

RSVP target-detection performance can be quantified by the subject's hit rate (HR) and false-alarm rate (FAR). Knowing whether a response is a hit or a false alarm requires knowing whether a target or a nontarget stimulus evoked the response. Because of response-time variability, it can be difficult to know what stimulus evoked a button-press response. For example, a response might be a relatively fast response to a target stimulus or a relatively slow response to the preceding nontarget stimulus. When the response-time variability substantially exceeds the interstimulus interval, situations arise in which a response could just as easily be attributed to any of several stimuli. One method currently in use for estimating HR and FAR entails establishing a temporal window after each target stimulus (e.g., 0–1 s relative to target onset) and declaring any response that falls in that window a hit. Other methods estimate a response-time probability density function and use that to assign responses to stimuli.

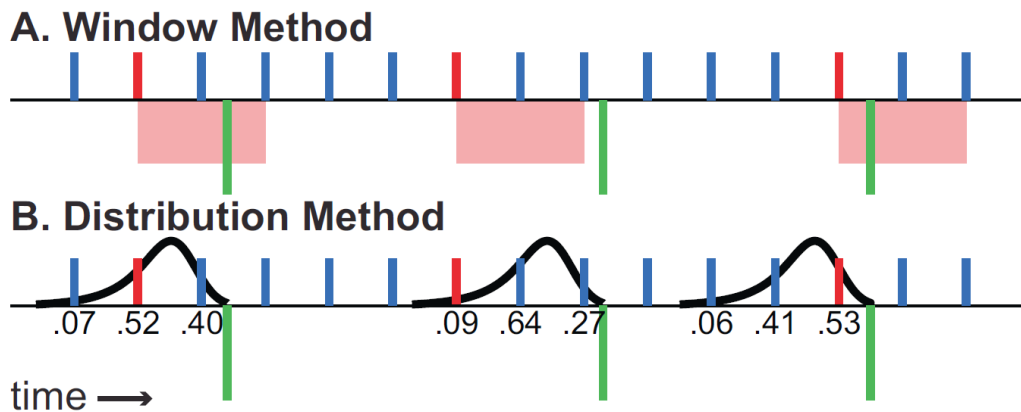
Here, a method is introduced that generally outperforms other methods currently in use for estimating the HR and FAR in RSVP target-detection tasks. This method and associated findings have been reported previously (Files and Marathe 2016). The purpose of this report is to provide instructions and source code for using this method while still providing context. Using simulations with known HRs and FARs, we show that the method introduced here is more accurate than established methods. This advantage is especially clear when the stimulus presentation rate is high and/or the FAR is nonzero. In addition to more accurately measuring the experimental effects of manipulations on target-detection performance, accurate estimates of detection performance can improve detection of target stimuli in applications in which the status of any given image as target or nontarget is unknown a priori.

### 3. Methods, Assumptions, and Procedures

#### 3.1 Estimation Methods

##### 3.1.1 Established Methods for Estimating HR and FAR

There are 2 classes of methods for determining HR and FAR in common use with RSVP target-detection tasks. The first class uses a windowing approach. This class of methods establishes a minimum and a maximum response time, typically from 0 to 1000 ms posttarget. Any response that falls within that window after a target is declared a hit, and then the HR is determined as the number of declared hits divided by the total number of targets. Responses that do not fall within a window corresponding to any target are declared false alarms, and the FAR is calculated as the number of false alarms divided by the number of nontarget stimuli (Fig. 1). Implementations of this method differ in how responses are scored when more than one response falls within a response window and/or what to do when a response falls within more than one response window.



**Fig. 1** Timelines illustrating existing response assignment methods. Blue hash marks indicate onset times of nontarget stimuli; red hash marks indicate onset times of target stimuli. Green hashes indicate times at which a response occurred. Interstimulus interval is 0.5 s. A) In the window method, a window of time (typically 0–1 s posttarget) is established. Responses falling within that window are declared hits. In this example, the first and third responses would be classified as hits, and the second would be classified as a false alarm. B) The same experiment timeline as analyzed with the distribution method. The black curves are the response time probability density function reversed and with its origin at the times of response. Numbers below stimulus hashes show the attribution resulting from the corresponding response, as computed using Eq. 1. Using maximum likelihood (the max method) assigns the response to the stimulus with the highest likelihood.

The second class of methods for estimating HR and FAR uses a response-time distribution to estimate a response-time probability density function (RT-PDF) that is used to assign responses to specific stimuli (Gerson et al. 2006). The likelihood

that a button press was in response to a specific candidate stimulus is estimated as the probability of that particular response time relative to the time of the candidate stimulus (i.e., the estimated value of the RT-PDF). The likelihood is then normalized by dividing the likelihood for each candidate stimulus by the sum of the likelihoods for all candidate stimuli (Marathe et al. 2014a). From here, the methods in this class diverge. One approach is to assign responsibility for the response to the stimulus with the maximum likelihood. If that stimulus is a target, the response is counted as a hit, and if the stimulus is a nontarget, the response is counted as a false alarm. The other approach is to distribute responsibility for the response to various stimuli according to the normalized likelihood that they generated the response. Because the distribution method is central to the method proposed in this report, it will be useful to define the function used to distribute responsibility, called here the apportionment function. Given times of stimulation  $S$ , a stimulus of interest at time  $S_i$ , a response at time  $T$ , and an RT-PDF function  $f$ , the apportionment function is defined as

$$A(S_i, T) = \frac{f(T-S_i)}{\sum_j f(T-S_j)}. \quad (1)$$

Using this approach, if the apportionment worked out such that 0.52 of the response was apportioned to a target stimulus and the remaining 0.48 was apportioned to a nontarget stimulus, then that response would count as 0.52 of a hit and 0.48 of a false alarm (Fig. 1).

### 3.1.2 Proposed Method

The regression method introduced here is based on the aforementioned apportionment method (Eq. 1). The proposed method estimates the expected response apportionment to each stimulus as a function of the probability that nearby stimuli will generate responses and the proportion of those possible responses that will be apportioned to the stimulus of interest. The expected response apportionment for the  $i$ th stimulus is the sum of the expected apportionment due to responses to all nearby stimuli,  $S_j$ :

$$E[A(S_i)] = \sum_j E[A_s(S_j, S_i)], \quad (2)$$

where  $A_s(S_j, S_i)$  is similar to  $A(S_i)$  but only computes the attribution onto  $S_i$  of responses actually generated by  $S_j$ . The expected value of  $A_s(S_j, S_i)$  is

$$E[A_s(S_j, S_i)] = \sum_T p(T) A(S_i, T). \quad (3)$$

The term  $p(T)$  is the probability that a response elicited by  $S_j$  occurs at time  $T$ . This term can be split into the probability that any response is elicited by stimulus  $S_j$ ,

denoted  $p(R|S_j)$ , times the probability that a response occurs at a specific time. This latter quantity is obtained from the response time probability density function,  $f$ .

$$E[A_s(S_j, S_i)] = p(R|S_j) \sum_T [f(S_j - T)A(S_i, T)]. \quad (4)$$

Substituting Eq. 4 into Eq. 2 yields the following:

$$E[A(S_i)] = \sum_j (p(R|S_j) \sum_T [f(S_j - T)A(S_i, T)]). \quad (5)$$

For simplicity of notation, the limits of summation for  $j$  and  $T$  are not given. However,  $f(x)$  is zero for negative  $x$  and approaches zero as  $x$  increases, and  $A(S_i, T)$  goes to zero as  $S_i - T$  increases in magnitude; so in practice, only a limited range of  $j$  and  $T$  need to be calculated.

This equation can be simplified under the assumptions of a typical RSVP target-detection experiment, namely that there are stimuli that are targets and stimuli that are nontargets. The probability of responding to a target is a constant hit rate  $HR$ , and the probability of responding to a nontarget is a constant false-alarm rate  $FAR$ . If the stimulus at  $S_j$  is a target,  $p(R|S_j, S_j \in tar)$  is  $HR$ . If the stimulus at  $S_j$  is a nontarget,  $p(R|S_j, S_j \in n.t.)$  is  $FAR$ . Separating out the target and nontarget stimuli near the stimulus of interest, the equation becomes

$$E[A(S_i)] = HR \times \sum_{S_j \in tar} \sum_T [f(S_j - T)A(S_i, T)] + FAR \times \sum_{S_j \in n.t.} \sum_T [f(S_j - T)A(S_i, T)]. \quad (6)$$

For each stimulus in the experiment, both summation terms can be computed based on the known stimulus timings and the RT-PDF. This yields a system of simple linear equations with one equation per stimulus and 2 unknowns:  $HR$  and  $FAR$ . Least-squares linear regression can then be used to find the values of  $HR$  and  $FAR$  that best fit the observed attribution for each stimulus; these are the estimates of the  $HR$  and  $FAR$  for the experiment.

### 3.1.3 Using the Proposed Method

An easy way to apply this method is to utilize the RSVP Performance Estimator (RPE) package provided under the Apache License 2.0 in the Appendix and available at the following GitHub page: <https://github.com/btfiles?tab=repositories>.

This package implements the regression method in MATLAB, using the statistics toolbox, and consists of 3 files: `RSVPPerformanceEstimator.m` (A.3), `fitExGauss.m` (A.4), and `exGaussPdf.m` (A.5) as well as a script illustrating their use (A.1).

To use the RPE package in a MATLAB script, set up 3 data variables, initialize the estimator, and run the estimator. The 3 data variables are *stim\_time*, *stim\_label*, and

*button\_time*. The *stim\_time* variable should be a vector of the times (in seconds) at which each stimulus presentation started. The *stim\_label* variable should be a vector of Boolean values designating each stimulus as a target (true) or nontarget (false). Both *stim\_time* and *stim\_label* should have length equal to the number of stimuli presented. The *button\_time* variable should be a vector of the times (in seconds) at which each button press started. With the 3 data variables set up, initializing the estimator requires the following line of code:

```
e = rpe.RSVPPerformanceEstimator(stim_time, stim_label, button_time);
```

Then, the estimator is run:

```
[hr, far] = e.runEstimates;
```

Estimator execution could take several minutes depending on the number and proximity (to each other in time) of the stimuli. Also, if multiple responses follow a stimulus, a warning is given, and only the first response is used. The estimator has 2 outputs: the estimated HR and the estimated FAR.

## 3.2 Evaluation Methods

---

Having introduced the mechanics of the proposed method, simulations are described that compare the performance of the proposed method with state-of-the-art methods. The general approach was to simulate responses based on a known HR and known FAR and then analyze the simulated data using the proposed method as well as the 3 other methods for estimating HR and FAR, as described (Fig. 2). To ensure that the stimulation timeline we used was well-founded, we used the timeline of stimulus and response events from a previously described RSVP target detection experiment (Marathe et al. 2013, 2014b; Ries and Larkin 2013). Portions of the methods of that experiment are summarized here because the stimulus timeline and response time distributions from that experiment were used in our simulations.

### 3.2.1 Participants

Fifteen participants (9 male, 6 female, ages 18–57, average 39.5) volunteered for the current study. Participants provided written informed consent, reported normal or corrected-to-normal vision, and reported no history of neurological problems. Fourteen of the 15 participants were right-handed. The voluntary, fully informed consent of the persons used in this research was obtained as required by federal and Army regulations (DOA 1990; DoD 1999). The investigator has adhered to Army policies for the protection of human subjects (DOA 1990).

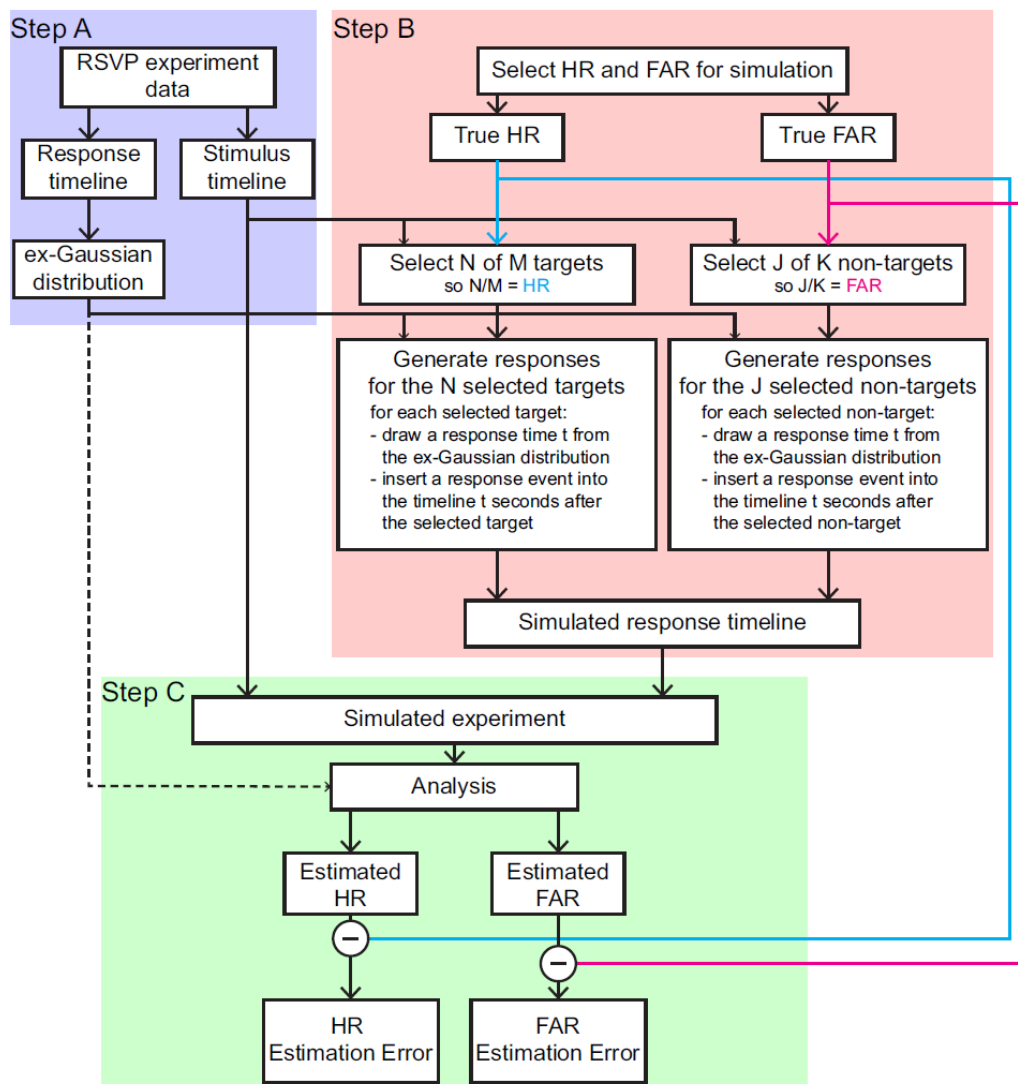
### **3.2.2 Stimuli and Procedure**

Stimuli consisted of short video clips that contained either people or vehicles in background scenes (target stimuli) or just background scenes (nontarget stimuli). Participants were instructed to make a manual button press with their dominant hand immediately when they detected a target and to abstain from responding to nontarget stimuli. Video clips consisted of 5 consecutive images, each 100 ms in duration; each video clip was presented for 500 ms. There was no interval between videos such that the first frame was presented immediately after the last frame of the prior video. If a target appeared in the video clip, it was present on each 100-ms image. The nontarget-to-target ratio was 90/10. RSVP sequences were presented in 2-min blocks, after which participants were given a short break. Participants completed a total of 10 blocks.

### **3.2.3 Simulations**

#### **3.2.3.1 Extracting a response-time probability density function**

All simulations and analyses were done using custom scripts in MATLAB version 2014a (MathWorks, Natick, Massachusetts). The RT-PDF used in the simulations was derived from the responses in the original timeline (Fig. 2, step A). An empirical response-time distribution was created by iterating over all target stimuli and looking for any response that fell between 200 and 1500 ms after the target. The latency of responses relative to the associated target events were then fit with an ex-Gaussian distribution using maximum-likelihood estimation (Lacouture and Cousineau 2008). The ex-Gaussian distribution is the sum of an exponential and a Gaussian; this distribution was selected because it compactly describes empirical response-time distributions reasonably well (Palmer et al. 2011). After estimating the RT-PDF, the responses in the original timeline were no longer considered for the simulations.



**Fig. 2 Simulation method; the process for one iteration of the simulation. It was repeated 250 times per combination of HR and FAR. Analysis was done separately using each of the 4 analysis methods described in the text.**

### 3.2.3.2 Simulating responses

Several simulations were then run to determine the accuracy with which the estimation methods described recover the HR and FAR under different true values of those quantities. A total of 101 HRs, ranging uniformly from 0 to 1, were combined with 101 FARs, also ranging uniformly from 0 to 1, resulting in 10,201 combinations of HR and FAR. To collect statistics on the performance at each combination of HR and FAR, each simulation was repeated 250 times.

For each simulation, an HR and an FAR were selected (Fig. 2, step B). Then a random subset of all targets and nontargets was selected to generate responses such

that the simulated rates were as close as possible to the selected rates (while still having whole numbers of responses). When a response was generated, a random draw was taken from the response-time distribution (as described by the RT-PDF), and a response event was added at that time after the generating stimulus.

### 3.2.3.3 Analyzing the simulated experiment

After simulating all of the responses necessary to generate the target HRs and FARs, the stimulus and simulated response timelines were analyzed using the 4 methods described: the window method, the maximum likelihood method (max), the distribution method, and the regression method (Fig. 2, Step C). Three stimulus presentation rates (stimuli per second) were simulated as well: 2, 4, and 10 Hz. The original experiment used a presentation rate of 2 Hz. To simulate faster presentation rates, the sampling rate of the experiment was multiplied by 2 and 5, respectively, while leaving the response-time distribution unchanged. This guaranteed that any change in the HR and FAR estimates was due to the presentation rate and not a difference in the total number of stimuli.

Three of the 4 methods tested (all but the window method) make use of the RT-PDF. In the first round of simulations, these 3 methods used the same RT-PDF that generated the data. In an experiment setting, however, the RT-PDF is not known a priori and must be estimated. When the HR is high enough and the FAR is low enough, an RT-PDF can be estimated from the data, as outlined previously. However, if the HR is suspected to be low, the method may produce an inaccurate estimate of the RT-PDF. We wanted to examine the relative performance of these methods when the RT-PDF cannot be estimated. In the second round of simulations, to simulate a worst-case scenario, the 3 methods that rely on an RT-PDF estimate were provided an RT-PDF that was uniform over the interval [0, 1000 ms]. That interval was chosen to correspond to the interval used by the window method. This flat RT-PDF introduces a high probability of multiple stimuli receiving equal attribution for a given response. This is relevant to the max method, because it assigns full attribution to the stimulus with maximal attribution. To resolve ties, the max method attributes the response to the earliest stimulus with maximal attribution.

Finally, to examine the impact that the choice of method for HR and FAR estimation can have on experimental results, the HR and FAR were estimated using the actual (rather than simulated) response data.

## 4. Results and Discussion

---

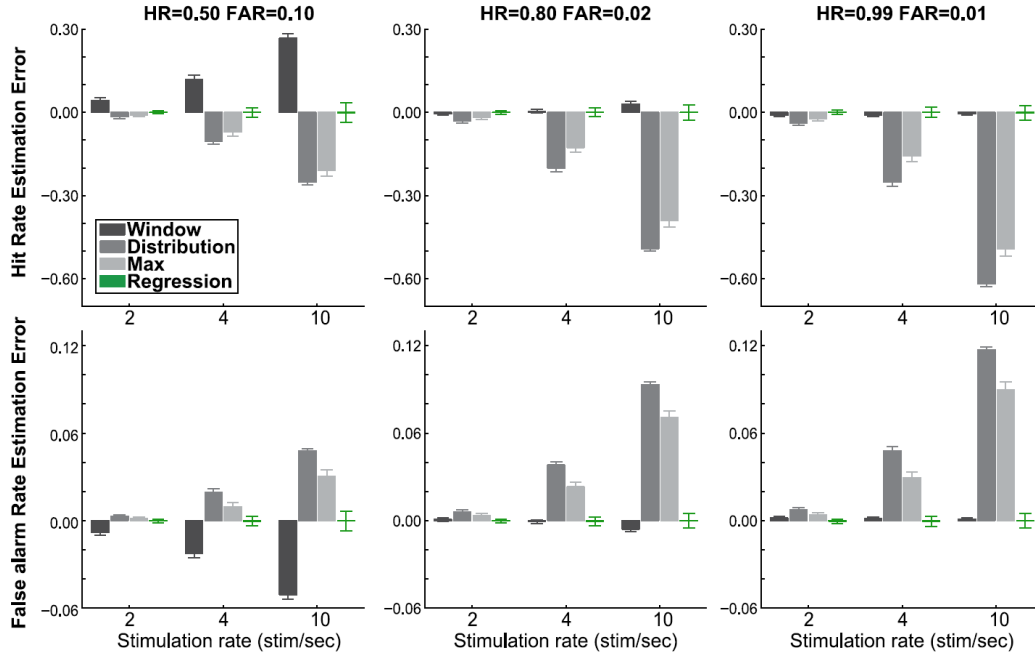
For each simulation, the HR and FAR estimation errors were computed as the difference between the simulated rate and the rate estimated by the estimation method under examination. For example, if the true HR was 0.8, but the method estimated the HR to be 0.75, the estimation error would be  $-0.05$ .

The remainder of this section is organized as follows. First, an illustrative subset of the simulation results is presented. This subset was chosen to show simulation results for HRs and FARs that might be obtained with poor, good, or excellent target-detection performance. Second, all of the simulation results are summarized to provide a comprehensive overview of the performance of the 4 estimation methods. Third, results of statistical tests are presented that tested for bias in the estimation methods used. Fourth, the results of simulations with an inaccurate RT-PDF are summarized. Finally, the results of applying each of the 4 estimation methods to real (rather than simulated) RSVP target-detection data are shown to illustrate the practical impact that the choice of estimation method can have.

### 4.1 Illustrative Subset of Results

---

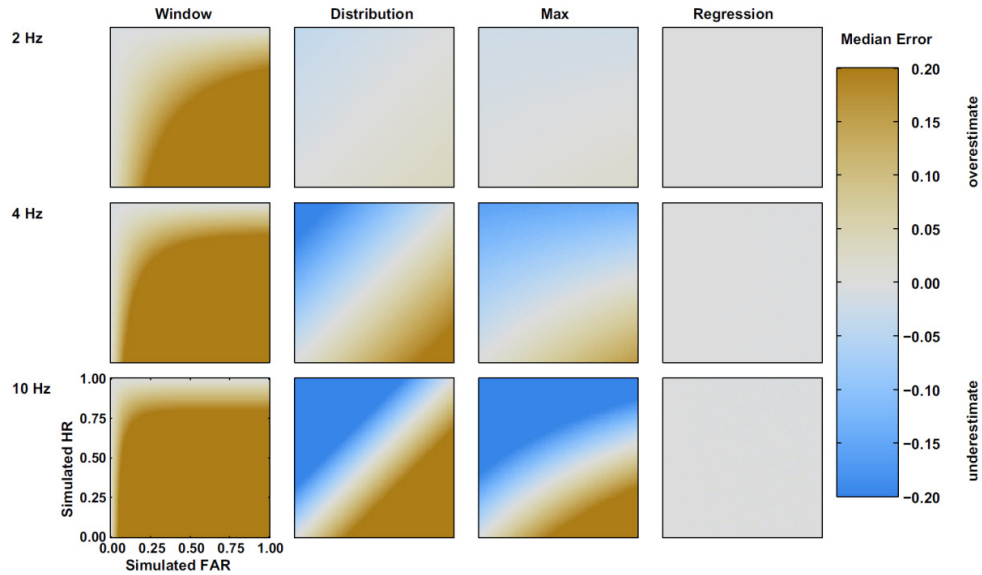
Although actual performance in RSVP target detection will depend heavily on the stimuli, task, and participant, 3 pairs of HR and FAR were chosen as illustrative exemplars of poor (HR 0.50, FAR 0.10), good (HR 0.80, FAR 0.02), and excellent (HR 0.99, FAR 0.01) performance (Fig. 3). Overall, when HR is high and FAR is low (i.e., in the good and excellent performances), the distribution and max methods make larger systematic errors than the other 2 methods, and the window method makes errors comparable to the regression method. As the presentation rate increases, the difference in the relative performance increases as well. In the poor performance case, the errors made by the regression method are clearly smaller than the others except at the 2-Hz presentation rate. At that rate, the regression, max, and distribution methods make comparable errors that are smaller than the errors made by the window method.



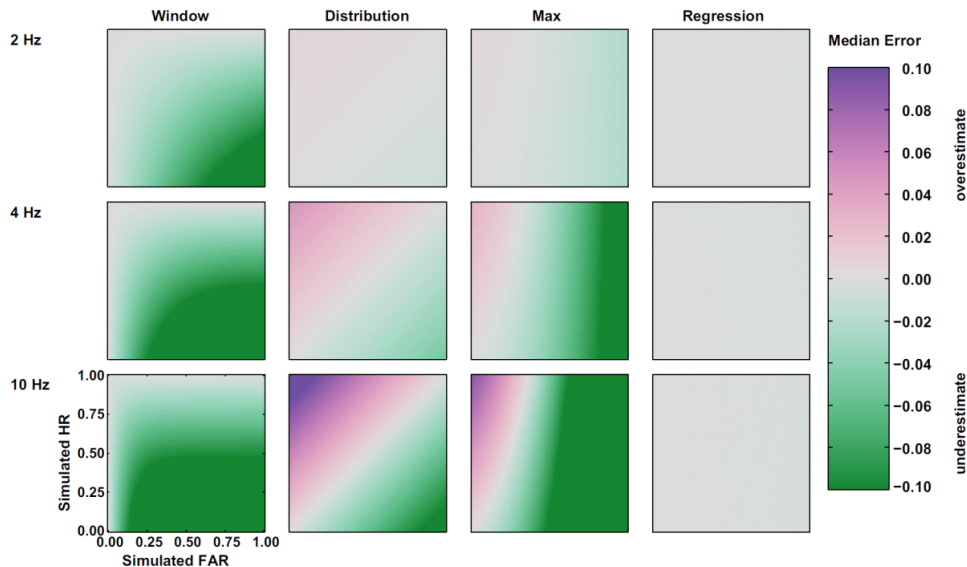
**Fig. 3** Estimation method performance examples. These plots illustrate estimation results for specific combinations of simulated HR and FAR when the true probability mass function of the response times was known. The pairs HR 0.50, FAR 0.10; HR 0.80, FAR 0.02; and HR 0.99, FAR 0.01 were selected as illustrative of poor, good, and excellent RSVP target-detection performance, respectively. Bars show the median estimation error, and error bars show plus/minus one standard deviation for each of the 4 estimation methods at 3 presentation rates. The upper row of panels show HR estimation errors, and the lower row shows FAR estimation errors.

## 4.2 Full Results

Considering the full range of simulated HRs and FARs, for all but the regression method, substantial systematic errors were apparent that depended on a combination of the simulated HR, simulated FAR, and simulated presentation rate for HR estimation (Fig. 4) and FAR estimation (Fig. 5). Estimation errors taken over the entire range of simulated HR and FAR were smallest for the regression method at all simulated rates with median absolute difference between estimated and simulated HRs of 0.001, 0.002, and 0.004 for presentation rates of 2, 4, and 10 Hz, respectively (Table 1), as well as median absolute difference for FARs of 0.001, 0.002, and 0.002 for presentation rates of 2, 4, and 10 Hz, respectively (Table 2). However, the regression method also had the largest variability for HR estimates at 4- and 10-Hz presentation and FAR at 10 Hz, measured as the standard deviation of all estimates after the median of all 250 estimates within a simulated HR/FAR cell had been subtracted (Tables 1 and 2). For the HR estimate, the regression method's variability was 0.022 and 0.053 at 4 and 10 Hz, respectively, and for the FAR estimate, the regression method's variability was 0.010 at 10 Hz.



**Fig. 4** HR estimation error summary. Each panel shows the simulation results for one of the estimation methods (columns) at a particular presentation rate (rows) when the true PDF of the response times was known. Colors indicate the difference between the median estimate of the HR and the simulated value of HR, clipped to an absolute value of 0.2. Within a panel, simulated FAR increases from left to right, and simulated HR increases from bottom to top. All methods except the regression method have HR estimation errors that clearly depend on the simulated HR and FAR, and the overall magnitude of errors increases as the presentation rate increases.



**Fig. 5** FAR estimation error summary. Each panel shows the simulation results for one of the estimation methods (columns) at a particular presentation rate (rows) when the true PDF of the response times was known. Colors indicate the difference between the median estimate of the FAR and the simulated value of FAR clipped to an absolute value of 0.1. Within a panel, simulated FAR increases from left to right, and simulated HR increases from bottom to top. All methods except the regression method have FAR estimation errors that clearly depend on the simulated HR and FAR, and the overall magnitude of errors increases as the presentation rate increases.

**Table 1 HR estimate performance with accurate RT-PDF**

Method	Presentation rate (Hz)					
	2		4		10	
	err	std	err	std	err	std
Window	0.280	0.016	0.447	0.012	0.518	0.008
Distribution	0.016	0.007	0.106	0.016	0.260	0.013
Max. attrib.	0.012	0.007	0.080	0.017	0.217	0.023
Regression	0.001	0.008	0.002	0.022	0.004	0.053

Notes: err = median absolute difference of each estimate from simulated values; std = standard deviation of estimates with median error subtracted.

**Table 2 FAR estimate performance with accurate RT-PDF**

Method	Presentation rate (Hz)					
	2		4		10	
	err	std	err	std	err	std
Window	0.053	0.003	0.085	0.002	0.099	0.002
Distribution	0.003	0.001	0.020	0.003	0.049	0.003
Max. attrib.	0.010	0.002	0.064	0.005	0.144	0.006
Regression	0.001	0.001	0.002	0.004	0.002	0.010

Notes: err = median absolute difference of each estimate from simulated values; std = standard deviation of estimates with median error subtracted.

### 4.3 Statistical Assessment of Bias in Estimation

To statistically assess the extent to which estimation errors depended on simulated HR, simulated FAR, and simulated presentation rate, HR estimation errors were first analyzed with a 4-way analysis of variance (ANOVA) with a categorical factor of estimation method (window, max, distribution, or regression) and continuous factors of presentation rate (2, 4, and 10 Hz), simulated HR, and simulated FAR (both ranging from 0 to 1.0 at 0.01 increments) (Table 3).

**Table 3 Omnibus ANOVA for HR estimation with accurate RT-PDF**

Source	d.f.	F	$\eta^2$	p-value
Method	3	$5.67 \times 10^5$	0.0198	0.0000
Rate	1	$3.19 \times 10^6$	0.0372	0.0000
HR	1	$7.47 \times 10^5$	0.0087	0.0000
FAR	1	$1.88 \times 10^6$	0.0218	0.0000
method*rate	3	$2.97 \times 10^6$	0.1039	0.0000
method*HR	3	$4.35 \times 10^6$	0.1520	0.0000
method*FAR	3	$2.16 \times 10^5$	0.0075	0.0000
rate*HR	1	$2.79 \times 10^6$	0.0324	0.0000
rate*FAR	1	$9.45 \times 10^6$	0.1100	0.0000
HR*FAR	1	$8.13 \times 10^5$	0.0094	0.0000
method*rate*HR	3	$1.86 \times 10^6$	0.0650	0.0000
method*rate*FAR	3	$1.08 \times 10^6$	0.0378	0.0000
method*HR*FAR	3	$9.97 \times 10^5$	0.0348	0.0000
rate*HR*FAR	1	$9.54 \times 10^3$	0.0001	0.0000
method*rate*HR*FAR	3	$7.96 \times 10^4$	0.0027	0.0000
Error	$3.06 \times 10^7$	...	0.3564	...

Because estimation method interacted with all other factors, individual ANOVAs were run for each method with factors presentation rate, simulated HR, and simulated FAR. Results of method-specific analyses are presented in Tables 4–7. In summary, all factors and interactions were statistically significant for the window method, with the 2 largest effects, measured with  $\eta^2$ , being the HR ( $\eta^2 = 0.226$ ) and presentation rate ( $\eta^2 = 0.225$ ). For the max method, all factors and interactions were statistically significant, with the interaction of presentation rate with HR ( $\eta^2 = 0.240$ ) and the interaction of presentation rate with FAR ( $\eta^2 = 0.355$ ) being the 2 largest effects. For the distribution method, all factors and interactions except the main effect of presentation rate and the interaction of HR with FAR were statistically significant, with the interaction of presentation rate with HR ( $\eta^2 = 0.370$ ) and of presentation rate with FAR ( $\eta^2 = 0.371$ ) having the largest effects. For the regression method, HR, FAR, and the interaction of those with presentation rate as well as the 3-way interaction were statistically significant, but the effect sizes of all factors and interactions were less than  $10^{-4}$ . This indicated that although the regression method's estimates do systematically depend on the presentation rate, HR, and FAR, the effects each account for less than 0.01% of the variance in the data. The statistical analysis on the FAR estimation errors produced similar results.

**Table 4 Method-specific ANOVA window method**

Source	d.f.	F	$\eta^2$	p-value
Rate	1	$4.34 \times 10^6$	0.2245	0.0000
HR	1	$4.37 \times 10^6$	0.2264	0.0000
FAR	1	$2.08 \times 10^5$	0.0108	0.0000
rate*HR	1	$1.28 \times 10^5$	0.0066	0.0000
rate*FAR	1	$1.25 \times 10^6$	0.0645	0.0000
HR*FAR	1	$1.36 \times 10^6$	0.0702	0.0000
rate*HR*FAR	1	$2.13 \times 10^4$	0.0011	0.0000
Error	$7.65 \times 10^6$	...	0.3960	...

**Table 5 Method-specific ANOVA max attribution method**

Source	d.f.	F	$\eta^2$	p-value
Rate	1	$2.61 \times 10^4$	0.0009	0.0000
HR	1	$1.53 \times 10^6$	0.0499	0.0000
FAR	1	$2.56 \times 10^6$	0.0838	0.0000
rate*HR	1	$7.34 \times 10^6$	0.2400	0.0000
rate*FAR	1	$1.09 \times 10^7$	0.3554	0.0000
HR*FAR	1	$5.83 \times 10^4$	0.0019	0.0000
rate*HR*FAR	1	$5.58 \times 10^5$	0.0182	0.0000
Error	$7.65 \times 10^6$	...	0.2500	...

**Table 6 Method-specific ANOVA distribution method**

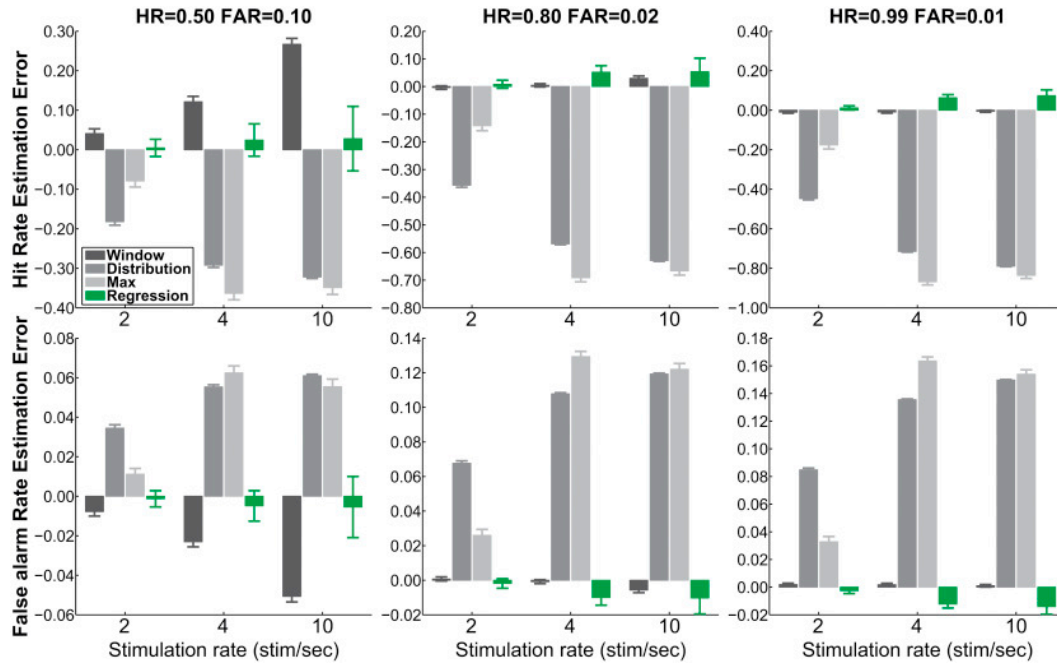
Source	d.f.	F	$\eta^2$	p-value
Rate	1	1.26	0.0000	0.2620
HR	1	$4.79 \times 10^6$	0.0718	0.0000
FAR	1	$4.81 \times 10^6$	0.0720	0.0000
rate*HR	1	$2.47 \times 10^7$	0.3704	0.0000
rate*FAR	1	$2.48 \times 10^7$	0.3711	0.0000
HR*FAR	1	0.11	0.0000	0.7396
rate*HR*FAR	1	6.71	0.0000	0.0096
Error	$7.65 \times 10^6$	...	0.1146	...

**Table 7 Method-specific ANOVA regression method**

Source	d.f.	F	$\eta^2$	p-value
Rate	1	0.00	0.0000	0.9565
HR	1	233.38	0.0000	0.0000
FAR	1	115.08	0.0000	0.0000
rate*HR	1	101.94	0.0000	0.0000
rate*FAR	1	190.05	0.0000	0.0000
HR*FAR	1	2.29	0.0000	0.1303
rate*HR*FAR	1	15.40	0.0000	0.0001
Error	$7.65 \times 10^6$	...	0.9999	...

#### 4.4 Simulations Run with Inaccurate RT-PDF Estimates

The second set of simulations used flat RT-PDF estimates to assess the performance of the RT-PDF-dependent methods when the estimated RT-PDF does not match the true RT-PDF. Results for HR and FAR estimation are detailed in Figs. 6–8.

**Fig. 6 Estimation method performance examples with flat RT-PDF**

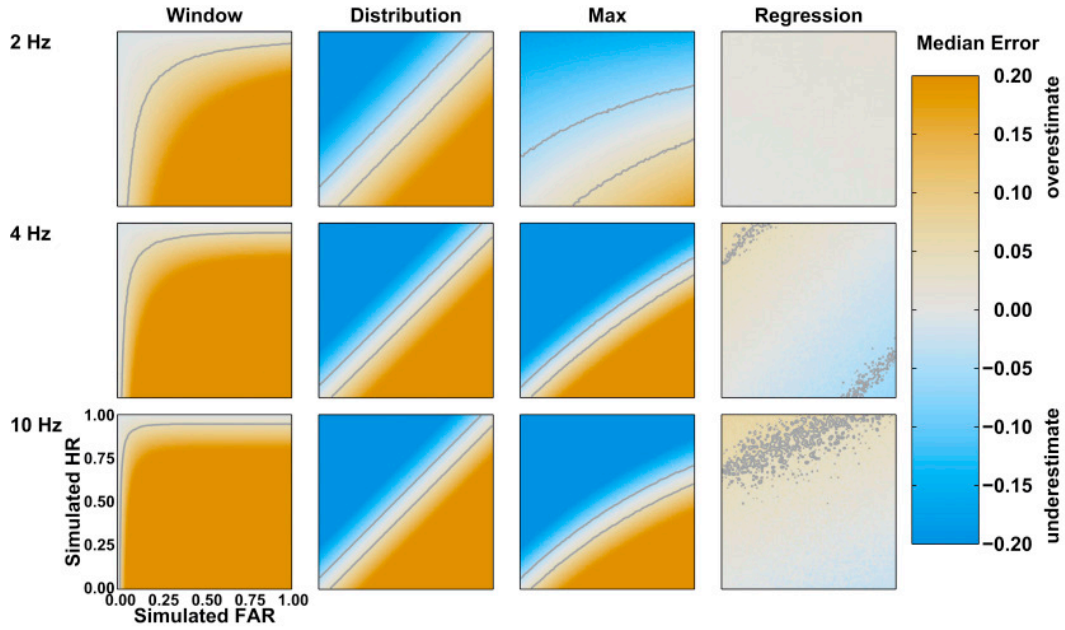


Fig. 7 HR estimation error summary with flat RT-PDF

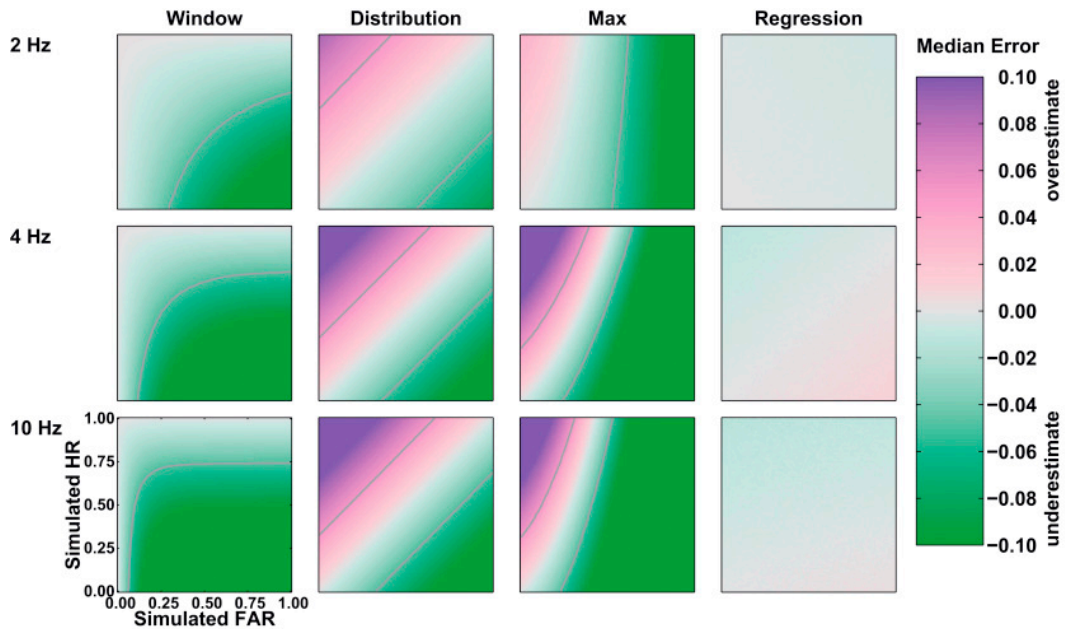


Fig. 8 FAR estimation error summary with flat RT-PDF

Numerical summaries for HR and FAR estimation are shown in Tables 8 and 9. Compared with the correct RT-PDF results, the distribution and max methods both had larger errors over a larger range of HR and FAR when using the flat RT-PDF. The regression method's estimation errors increased somewhat by using the incorrect RT-PDF, but overall errors were smallest.

**Table 8 HR estimate performance with flat RT-PDF**

Method	Presentation rate (Hz)					
	2		4		10	
	err	std	err	std	err	std
Window	0.280	0.016	0.447	0.012	0.518	0.008
Distribution	0.190	0.014	0.299	0.009	0.333	0.005
Max. attrib.	0.088	0.017	0.347	0.022	0.328	0.023
Regression	0.010	0.030	0.027	0.058	0.033	0.105

Notes: err = median absolute difference of each estimate from simulated values; std = standard deviation of estimates with median error subtracted.

**Table 9 FAR estimate performance with flat RT-PDF**

Method	Presentation rate (Hz)					
	2		4		10	
	err	std	err	std	err	std
Window	0.053	0.003	0.086	0.002	0.099	0.001
Distribution	0.037	0.003	0.059	0.002	0.065	0.001
Max. attrib.	0.079	0.005	0.132	0.006	0.176	0.006
Regression	0.003	0.006	0.004	0.011	0.004	0.020

Notes: err = median absolute difference of each estimate from simulated values; std = standard deviation of estimates with median error subtracted.

## 4.5 Analyzing Experiment Data

As an example of the effect of using different analysis methods on real data, the actual (rather than simulated) responses were analyzed. HR estimates are shown for each of the 15 subjects in Fig. 9. HRs were fairly high, ranging from 78.4% to 90.5% across subjects and estimation methods. A one-way repeated-measures ANOVA revealed a significant effect of analysis method on HR estimate ( $F(3,42) = 36.0$ ,  $p = 1.1 \times 10^{-11}$ ,  $\eta^2 = 0.131$ ). Follow-up paired comparisons (Bonferroni corrected) showed that the distribution estimate ( $M = 0.843$ ,  $SE = 0.002$ ) and max estimate ( $M = 0.848$ ,  $SE = 0.002$ ) were not significantly different ( $p = 0.25$ ), and the window ( $M = 0.864$ ,  $SE = 0.002$ ) and regression ( $M = 0.864$ ,  $SE = 0.002$ ) estimates were also not significantly different ( $p = 1.0$ ), but both max and distribution estimates were significantly lower than both the window and regression estimates (all  $p < 0.00001$ ).

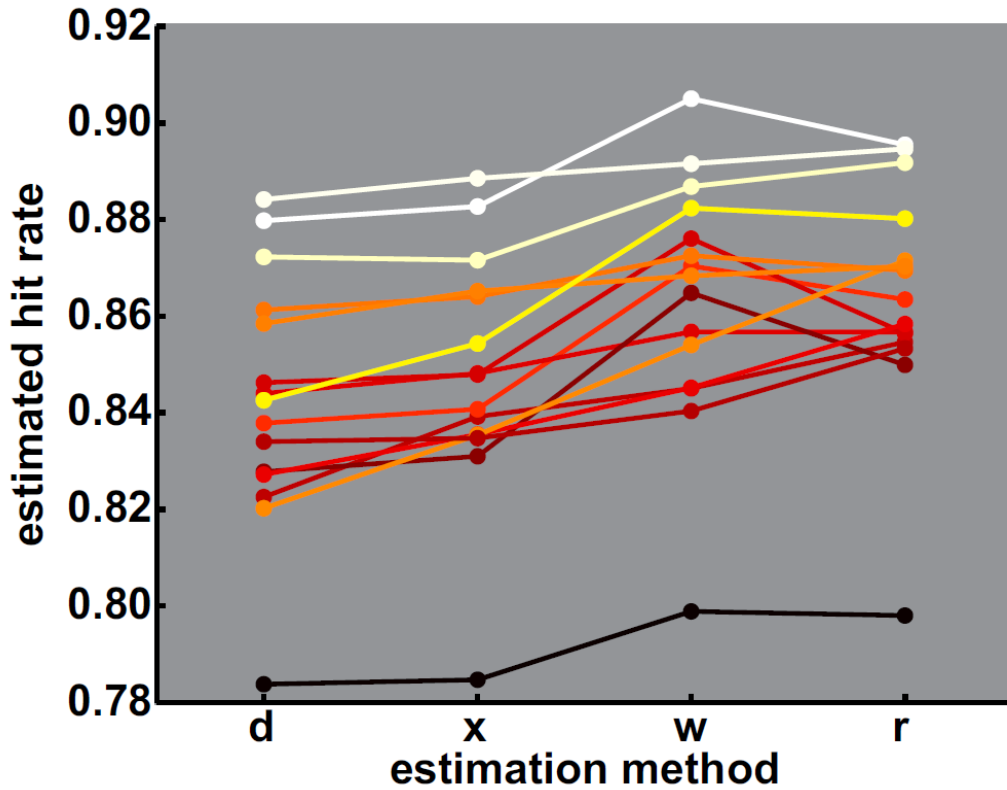


Fig. 9 Estimated HRs from experimental data with 4 different estimation methods. HR was estimated from the response data from 15 subjects using the distribution (d), max (x), window (w), and regression (r) methods. The colors for individual subject data are based on the estimates from the regression method to illustrate how the relative ordering of subjects changes based on estimation method.

#### 4.6 Discussion

The primary goal of these simulations was to test how well the proposed regression method for estimating HR and FAR in RSVP target-detection tasks could recover the true simulated HR and FAR relative to established methods. The simulation results showed that the proposed regression method was more accurate than established methods, although accuracy comes at the cost of some precision.

The simulations comparing the performance of the 4 HR and FAR estimators revealed systematic errors in all 4 methods, such that the error in HR and FAR estimates depended on some combination of the true value of the HR, FAR, and presentation rate, but the inaccuracy of the 4 methods was not equivalent.

The window method overestimates the HR as the true HR decreases and/or the true FAR increases. This can be understood as the result of the benefit-of-the-doubt approach this method represents. Any response within a window of a target is declared a hit by this method, so any false alarm that occurs in temporal proximity

to a target might be incorrectly classified as a hit. Additionally, responses to targets that are slow enough to fall outside the 1-s window will be misclassified as misses. An important property of the window method demonstrated in the results here is that when the true FAR is very low, this method yields fairly accurate estimates of HR and FAR. This is because as the FAR approaches zero, the vast majority of responses will actually be hits, and the vast majority of hits should fall within the window and therefore be correctly classified by this method. This was illustrated in the “excellent” performance simulation (Fig. 3) in which the HR was slightly underestimated and the FAR was slightly overestimated.

Overall, the max and distribution methods made smaller errors in HR estimation than the window method (Table 1), although errors were relatively large in the range of HR and FAR that might be associated with good or excellent task performance (Fig. 3). Both of these methods had their lowest estimation errors when the simulated HR and FAR were similar. Because RSVP experiments typically report fairly high HR and low FAR, in practice, both of these methods are expected to underestimate the HR and overestimate the FAR.

The regression method had lower estimation error than the other 3 methods, and the errors do not depend strongly on the true values of HR and FAR. The distribution method makes systematic errors that depend strongly on the true HR, FAR, and presentation rate (Table 4), and the regression method attempts to correct for those errors by accounting for how errors contribute to the expected value of the apportionment to any given stimulus using linear regression. The statistical analysis of the estimation errors of the regression method revealed a reliable effect of the interaction of FAR with presentation rate, but the effect size was less than  $10^{-4}$ . The absence of nontrivial linear effects revealed in the ANOVA is evidence that the linear regression method accomplished its goal. Nonlinear effects could potentially affect the estimation error of the regression method, but given the small overall estimation error of the regression method (Tables 1 and 2), any such effects do not appear to have a major impact, at least under the conditions simulated here.

The presentation rate had a sizeable impact on estimate accuracy on all of the estimation methods except the regression method, although the precision of the regression method’s estimates decreased as the presentation rate increased. The increases in estimation error can be understood as a consequence of the increasing ambiguity of which stimulus elicited a particular response. Although such a slow rate was not tested here, clearly if the stimuli are spaced far enough apart, then errors in response assignment would be very rare. As more stimuli fall into a temporal range of plausibly causing a response, the harder it will be to correctly assign that response to a stimulus.

One potential caveat to the apparent success of the regression method is that in our simulations, the regression method was provided with the true probability density function for response times (RT-PDF). In practical use, the RT-PDF would have to be estimated from the available data. For completeness, simulations included true HRs that were low or zero. In those situations, estimating an RT-PDF would be difficult or impossible, so in our second set of simulations we provided all of the analysis methods with a highly incorrect, uniform RT-PDF (Tables 5 and 6). Having such a poor estimate of the RT-PDF did not obliterate the RT-PDF-dependent methods, although the performance of those methods did drop somewhat. Based on this result, it seems that even if estimation of the RT-PDF is poor, the regression method may still be recommended.

An assumption of the regression method is that responses to different stimuli are independent. Strictly, this assumption is incorrect for 2 reasons. First, the method assumes that it is possible for 2 responses to occur at the same time (e.g., a slow response to an earlier stimulus occurs simultaneously with a fast response to a later stimulus), but in practice there are limits to how quickly a person can press a button twice. This first assumption was in fact violated in the simulations run here, because in the rare event that multiple responses occurred at the same time, those responses were conflated into a single response. The chance of response collisions increases as the number of overall responses increases, and this would be most prevalent at fast presentation rates with high FARs, and it might explain the small but significant interaction of FAR with presentation rate that impacted the estimation error of the regression method.

Second, humans typically fail to perceive images that fall within a short window of time starting shortly after a target image. This phenomenon is called the attentional blink (Raymond et al. 1992; Shapiro et al. 1997). This could temporarily lower the HR and/or the FAR by reducing the probability of responding for a short time after each response. There was no modeling of the attentional blink in the simulations done here, so its impact on any of the estimation methods here cannot be assessed.

To illustrate the impact the choice of HR/FAR estimation method might have in an experimental setting, behavioral results from a target-detection experiment were analyzed using the 4 methods tested in simulations. The impact of analysis method on the overall HR and FAR estimates was statistically significant, and the effect of analysis method was consistent with the simulation results of good performance overall. Qualitatively, this provides support for the validity of our simulations. However, for some individuals, the regression method estimated a somewhat higher HR than the window method (Fig. 9). Inspection of the responses from the subjects for whom the regression method had a higher estimate than the window method revealed that these subjects appeared to occasionally respond twice within a

500-ms span (corresponding to the interstimulus interval). If a single target image elicits 2 responses, the window method calls one a hit and the other a false alarm, so double-responding would not inflate the HR estimate. The regression method, however, does not have special handling of double responses, and they could inflate the HR estimate. Based on these data, we cannot know if these responses are examples of nonindependence. It could be that the subjects inadvertently pressed the response button twice after seeing a target image, or it could be that the 2 button presses were intended as responses to consecutive images.

## 5. Conclusions

---

Based on its better estimation of HR and FAR, the regression method proposed here would seem the best choice when estimating the HR and FAR is the primary interest. If the FAR is known to be essentially equal to 0, the window method may have an advantage because the window method is somewhat simpler to implement and is more precise with faster presentation rates. In the more general case in which the FAR may be non-negligible and a fast presentation rate is used, the regression method is likely to provide the most-accurate estimates of HR and FAR.

In real-world applications, the goal of using an RSVP target-detection paradigm may not be to estimate the HR and FAR but to find targets in a set of images when it is unknown whether any particular image constitutes a target. When the status of an image as a target is unknown, the window and regression methods cannot be applied directly, so alternative methods are needed. Both the distribution and max methods can be applied to unknown images, but when the human operator's performance is good, these methods have poor performance in the aggregate. Past efforts have used a Bayesian formulation to estimate the probability that a stimulus is a target given a response at some latency relative to the target (Gerson et al. 2006). That method includes estimated HR and FAR terms that must be learned from some training data set. With the more-accurate HR and FAR estimates afforded by the regression method proposed here, more-accurate estimates of target probability can also be derived.

Although the focus of this report is on target-detection accuracy in the RSVP paradigm, many related projects focus on using some physiological measure to enable a brain-computer interface for target detection (Gerson et al. 2006; Luo and Sajda 2009; Privitera et al. 2010; Sajda et al. 2010). Electroencephalography (EEG)-based classification can sometimes classify images correctly even when the behavioral response was incorrect (Sajda et al. 2003; Bigdely-Shamlo et al. 2008).

Brain-activity-based classification may be less susceptible to temporal uncertainty, because sensory processing is less temporally variable than behavioral responses

(Schall and Bichot 1998). However, temporal variability remains in neural responses used for classification with EEG. Classification methods are in use that are robust to temporal variability of neural signals (Rivet et al. 2009; Marathe et al. 2014b), but the same ambiguity in mapping button-press responses back to their evoking stimuli applies when mapping event-related potential events back to their evoking stimuli. Approaches derived from the regression method introduced here should aid in resolving the ambiguity in assigning classification scores to appropriate stimulus images.

While the goal of using EEG or pupilometry instead of button presses to find targets in image databases holds promise, perhaps more promising is a fusion approach (Luo and Sajda 2006; Marathe et al. 2014a) in which both physiological and overt behavioral responses are combined. In such an approach, better estimates of behavioral characteristics derived from the regression method introduced here should lead to more-effective systems overall.

## 6. References

---

- Bigdely-Shamlo N, Vankov A, Ramirez RR, Makeig S. Brain activity-based image classification from rapid serial visual presentation. *IEEE Trans Neural Syst Rehabil Eng.* 2008;16(5):432–441.
- Chun MM, Potter MC. A two-stage model for multiple target detection in rapid serial visual presentation. *J Exp Psychol Hum Percept Perform.* 1995;21(1):109–127.
- [DOA] Department of the Army (US). Use of volunteers as subjects of research. Washington (DC): Government Printing Office (US); 1990. Army Regulation No.: AR 70-25.
- [DoD] Department of Defense (US), Office of the Secretary of Defense. Protection of human subjects. Washington (DC): Government Printing Office (US); 1999. Code of Federal Regulation No.: 32 CFR 219.
- Files BT, Marathe AR. A regression method for estimating performance in a rapid serial visual presentation target detection task. *J Neurosci Methods.* 2016;258(30):114–123.
- Gerson AD, Parra LC, Sajda P. Cortically coupled computer vision for rapid image search. *IEEE Trans Neural Syst Rehabil Eng.* 2006;14(2):174–179.
- Intraub H. Rapid conceptual identification of sequentially presented pictures. *J Exp Psychol Hum Percept Perform.* 1981;7(3):604–610.
- Kanwisher NG. Repetition blindness: type recognition without token individuation. *Cognition.* 1987;27(2):117–143.
- Keyesers C, Xiao D-K, Foldiak P, Perrett DI. The speed of sight. *J Cogn Neurosci.* 2001;13(1):90–101.
- Lacouture Y, Cousineau D. How to use MATLAB to fit the ex-Gaussian and other probability functions to a distribution of response times. *Tutor Quant Methods Psychol.* 2008;4(1):35–45.
- Luo A, Sajda P. Using single-trial EEG to estimate the timing of target onset during rapid serial visual presentation. *Proceedings of the IEEE Engineering in Medicine and Biology Society Conference;* 2006. Vol. 1. p. 79–82.
- Luo A, Sajda P. Comparing neural correlates of visual target detection in serial visual presentations having different temporal correlations. *Front Hum Neurosci.* 2009;3(5).

- Marathe AR, Ries AJ, McDowell K. A novel method for single-trial classification in the face of temporal variability. In: Schmorow DD, Fidopiastis CM, editors. *Foundations of augmented cognition*. Berlin (Germany): Springer; 2013. p. 345–352.
- Marathe AR, Lance BJ, Nothwang W, Metcalfe JS, McDowell K. Confidence metrics improve human-autonomy integration. *Proceedings of the 9th ACM-IEEE International Conference for Human–Robot Interaction*; 2014a Mar 3–6; Bielefeld, Germany.
- Marathe AR, Ries AJ, McDowell K, Sliding HDCA. Single-trial EEG classification to overcome and quantify temporal variability. *IEEE Trans Neural Syst Rehabil Eng*. 2014b;22(2):201–211.
- Mathan S, Ververs P, Dorneich M, Whitlow S, Carciofini J, Erdogmus D, Pavel M, Huang C, Lan T, Adami A. Neurotechnology for image analysis: searching for needles in haystacks efficiently. In: Schmorow DD, Stanney KM, Reeves LM, editors. *Augmented cognition: past, present, and future*. 2nd ed. Arlington (VA): Strategic Analysis, Inc.; 2006.
- Nasanen R, Ojanpaa H, Tanskanen T, Paallysaho J. Estimation of temporal resolution of object identification in human vision. *Exp Brain Res*. 2006;172(4):464–471.
- Palmer EM, Horowitz TS, Torralba A, Wolfe JM. What are the shapes of response time distributions in visual search? *J Exp Psychol Hum Percept Perform*. 2011;37(1):58–71.
- Parra LC, Christoforou C, Gerson AD, Dyrholm M, Luo A, Wagner M, Philiastides MG, Sajda P. Spatiotemporal linear decoding of brain state. *IEEE Signal Processing Magazine*. 2008;25(1):107–115.
- Potter MC, Levy EI. Recognition memory for a rapid sequence of pictures. *J Exp Psychol*. 1969;81(1):10–15.
- Privitera CM, Renninger LW, Carney T, Klein S, Aguilar M. Pupil dilation during visual target detection. *J Vis*. 2010;10(10):3.
- Raymond JE, Shapiro KL, Arnell KM. Temporary suppression of visual processing in an RSVP task: an attentional blink? *J Exp Psychol Hum Percept Perform*. 1992;18(3):849–860.
- Ries AJ, Larkin GB. Stimulus and response-locked P3 activity in a dynamic rapid serial visual presentation (RSVP) task. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2013 Jan. Report No.: ARL-TR-6314.

- Rivet B, Souloumiac A, Attina V, Gibert G. xDAWN algorithm to enhance evoked potentials: application to brain–computer interface. *IEEE Trans Biomed Eng.* 2009;56(8):2035–2043.
- Sajda P, Gerson A, Parra L. High-throughput image search via single-trial event detection in a rapid serial visual presentation task. *Proceedings of the 1st IEEE Engineering in Medicine and Biology Society Neural Engineering Conference*; 2003 Mar. p. 7–10.
- Sajda P, Pohlmeier E, Wang J, Parra LC, Christoforou C, Dmochowski J, Hanna B, Bahlmann C, Singh MK, Chang S-F. In a blink of an eye and a switch of a transistor: cortically coupled computer vision. *Proc IEEE.* 2010;98(3):462–478.
- Schall JD, Bichot NP. Neural correlates of visual and motor decision processes. *Curr Opin Neurobiol.* 1998;8(2):211–217.
- Shapiro KL, Raymond JE, Arnell KM. The attentional blink. *Trends Cogn Sci.* 1997;1(8):291296.

## **Appendix. Rapid Serial Visual Presentation Performance (RSVP) Estimator (RPE) Package Source Code**

---

## A.1 Licensing

Copyright 2016 Benjamin T. Files

Licensed under the Apache License, Version 2.0 (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## A.2 example\_script.m

```
%% Overview
% This script shows an example of how to use the
% RSVPPerformanceEstimator.
%
% The bulk of the script simulates an RSVP target detection
% experiment. To use this code with your data, you need to setup
% the following three variables:
% stim_time: nstim-length vector of stimulus times, in seconds
% with whatever precision is available, at least millisecond.
% stim_label: nstim-length vector of stimulus labels, true for
% target, false otherwise.
% button_time: The time (again, in seconds) of button press
% starts.
%
% Then initialize the estimator like so:
% e = rpe.RSVPPerformanceEstimator(stim_time, stim_label,
% button_time);
% and run the estimator:
% [hr, far] = e.runEstimates;
%
% As written, this example script simulates data and then
% estimates performance using the conventional window method and
% the regression method of Files & Marathe (2016). HR and FAR
% estimates are printed to the command window, and response time
% PDF and estimates are plotted.
%
% See Also rpe.RSVPPerformanceEstimator rpe.exGaussPdf
% rpe.fitExGauss
%
% Reference
% Files, B. T., & Marathe, A. R. (2016). A regression method for
% estimating performance in a rapid serial visual presentation
% target detection task. Journal of Neuroscience Methods, 258,
% 114?123. http://doi.org/10.1016/j.jneumeth.2015.11.003

%% Simulation settings
% This simulates an RSVP experiment. Stimuli are shown in
% blocks, with some amount of time between blocks. Stimulation
% rate, block length, inter-block interval and number of blocks
% are all configurable.
%
```

Approved for public release; distribution is unlimited.

```

% Some stimuli are assumed to be nontargets, and others are
% targets. The proportion of targets is configurable.
%
% For responses, a proportion of targets (pHit) generate a
% response and a proportion of nontargets (pFa) also generate a
% response. Response latencies are sampled from an exGaussian
% distribution configurable parameters.

% for deterministic performance
rng('default');
% change to
rng('shuffle')
% if you want to get a sense of how much variability you can get
% on repeated runs.

% stimulation settings
stim_rate = 4; % Stim/s
block_length = 60; % s
inter_block_interval = 10; % s
n_block = 10; % number of blocks
pTar = 0.1; % proportion of stimuli that are targets

% True performance parameters
pHit = 0.6; % hit rate
pFa = 0.1; % false-alarm rate

% exgaussian RT parameters
mu = 0.3;
sigma = 0.1;
tau = 0.15;

% exgaussian random numbers
exgr = @(sz) normrnd(mu, sigma, sz) + exprnd(tau, sz);

%% Run the simulation
% Setup stimulus times
block_stim = 0:(1/stim_rate):block_length;
stim_time_mtx = repmat(block_stim(:), 1, n_block);
blk_add = (0:(n_block-1)).*(block_stim(end) + inter_block_interval);
stim_time_mtx = bsxfun(@plus, blk_add, stim_time_mtx);
stim_time = stim_time_mtx(:)';

% setup stimulus labels
nTar = round(numel(stim_time)*pTar);
lbl = false(size(stim_time));
lbl(1:nTar) = true;
stim_label = lbl(randperm(numel(stim_time)));

% setup buttonpresses
nHit = round(pHit*sum(stim_label));
tar_times = stim_time(stim_label);
hit_idx = false(size(tar_times));
hit_idx(1:nHit) = true;
hit_idx = hit_idx(randperm(numel(hit_idx)));
hit_times = tar_times(hit_idx);
hit_responses = exgr(size(hit_times)) + hit_times;

nFa = round(pFa*sum(~stim_label));
nt_times = stim_time(~stim_label);
fa_idx = false(size(nt_times));
fa_idx(1:nFa) = true;

```

```

fa_idx = fa_idx(randperm(numel(fa_idx)));
fa_times = nt_times(fa_idx);
fa_responses = exgr(size(fa_times)) + fa_times;

button_time = sort([hit_responses fa_responses]);

%% Visualize the experiment timeline
% % Not a great visualization, but might be good for debugging
% figure('Name', 'Simulated Timeline');
% stem(tar_times, ones(size(tar_times)), 'b', 'marker', 'none');
% hold on;
% stem(nt_times, ones(size(nt_times)), 'g', 'marker', 'none');
% stem(button_time, 1.1 * ones(size(button_time)), 'k', 'marker',
% 'none');

%% Do a conventional window analysis
win_lo = 0.0;
win_hi = 1.0;
in_any_win = false(size(button_time));
n_hit = 0;
for iTar = 1:numel(tar_times)
    tt = tar_times(iTar);
    in_win = button_time > tt + win_lo & button_time < tt+win_hi;
    if any(in_win),
        n_hit = n_hit + 1;
    end
    in_any_win(in_win) = true;
end

win_hr = n_hit/numel(tar_times);
win_far = sum(~in_any_win)/numel(nt_times);

%% Do the regression estimation
% setup the estimator
e = rpe.RSVPPerformanceEstimator(stim_time, stim_label, button_time);

% Now run the estimator
tic
[hr, far] = e.runEstimates();
toc
% This step might take a few minutes, depending on how many
% stimuli you give it and how close the stimuli are together in
% time. Also, responses that are very close together in time are
% potentially problematic. In this example, a warning is thrown
% but otherwise ignored.

%% Report estimates
fprintf(1, '\n=== Window Method ===\n');
fprintf(1, 'Estimated HR \t%0.4f, true value was %g\n', win_hr, pHit);
fprintf(1, 'Estimated FAR \t%0.4f, true value was %g\n', win_far, pFa);

fprintf(1, '\n=== Regression Method ===\n');
fprintf(1, 'Estimated HR \t%0.4f, true value was %g\n', hr, pHit);
fprintf(1, 'Estimated FAR \t%0.4f, true value was %g\n', far, pFa);

%% Visualize the response time distributions
figure('Name', 'Simulated response distribution');
subplot(2,1,1);
edg = 0:.05:1.5;
N = histc([fa_responses-fa_times hit_responses-hit_times], edg);

```

```

bar(edg, N, 'histc');
x = 0:.001:1.5;
pdf = rpe.exGaussPdf(x, mu, sigma, tau);
npdf = numel(button_time)*pdf*diff(edg([1 2]));
hold on;
plot(x, npdf, 'r');
title('true distribution')

subplot(2,1,2);
eN = histc(e.rt_list, edg);
epdf = e.pdf_fcn(x);
enpdf = numel(e.rt_list)*epdf*diff(edg([1 2]));
bar(edg, eN, 'histc');
hold on;
plot(x, enpdf, 'r');
title('estimated distribution');

```

### A.3 RSVPPerformanceEstimator.m

```

classdef RSVPPerformanceEstimator < handle
% An implementation of a regression-based method for estimating
% hit rate and false-alarm rate in an RSVP target detection
% experiment.
%
% Justification, derivation, and simulations validating this
% method are presented in Files & Marathe, 2016.
%
% Methods will attempt to use parallel for (parfor) loops if the
% parallel processing toolbox function gcp() is available and
% returns without error.
%
% Example
% e = rpe.RSVPPerformanceEstimator(stim_time, stim_lbl, button_time);
% [hr, far] = e.runEstimates();
%
% Reference
% Files, B. T., & Marathe, A. R. (2016). A regression method for
% estimating performance in a rapid serial visual presentation
% target detection task. Journal of Neuroscience Methods, 258,
% 114?123. http://doi.org/10.1016/j.jneumeth.2015.11.003
%
% RSVPPerformanceEstimator properties:
% Must be set by user (in constructor)
% stimulus_times - times (s) at which stimuli were presented
% stimulus_labels - true for targets, false otherwise
% buttonpress_times - times (s) at which button was pressed
%
% Options with default values
% time_resolution - resolution of PDF approximation (s)
%                   default .001.
% response_window - response time window for RT estimates (s)
%                   default [0.0 1.0].
% pdf_support - how long after the stimulus to compute RT PDF (s)
%               default 1.5
%
% RSVPPerformanceEstimator methods:
% RSVPPerformanceEstimator - Constructor takes 3 arguments:
%                             stim_times, stim_lbls, button_times
% runEstimates - Estimates the response time PDF and uses that
%               to estimate HR and FAR. Uses estimatePdf and
%               estimatePerformance

```

```

%
% See Also example_script

properties
    stimulus_times; % times (s) at which stimuli were presented
    stimulus_labels; % true for targets, false otherwise
    buttonpress_times; % times (s) at which button was pressed

    % configurable parameters
    time_resolution = 0.001; % resolution of PDF approximation
    % response time window for RT estimates
    response_window = [0.0 1.0];
    % how long after the stimulus to compute RT PDF
    pdf_support = 1.5;
end

properties (GetAccess=public, SetAccess=private)
    % estimated parameters of the exGaussian response time
    % distribution
    mu % mean of the gaussian
    sigma % standard deviation of the gaussian
    tau % parameter of the exponential

    % best guess at collection of response times
    rt_list

    % rt pdf convenience values:

    pdf_fcn % function handle for the PDF estimate
    pdf_est % pre-computed PDF values

    beta % explanatory variable for regression
    response_scores % dependent variable for regression
end

methods
function obj = RSVPPerformanceEstimator(varargin)
    % Takes 3 arguments: stim_time, stim_lbl,
    % buttonpress_time.
    if nargin == 0,
        return
    end

    ip = inputParser();
    ip.addRequired('stim_t');
    ip.addRequired('stim_lbl');
    ip.addRequired('bp_t');

    ip.parse(varargin{:});
    obj.stimulus_times = ip.Results.stim_t;
    obj.stimulus_labels = ip.Results.stim_lbl;
    obj.buttonpress_times = ip.Results.bp_t;
end
function [hr, far, hrci, farci] = runEstimates(obj, cialpha)
    % Estimate the rt pdf, HR and FAR.
    % [hr, far] = updateEstimates()
    %
    % See also estimatePdf estimatePerformance

```

```

    if nargin < 2,
        cialpha = .05;
    end

    obj.estimatePdf;
    if nargin == 2,
        [hr, far] = obj.estimatePerformance;
    elseif nargin == 4,
        [hr, far, hrci, farci] = obj.estimatePerformance(cialpha);
    end
end

function estimatePdf(obj)
    % estimates a new response time PDF
    % Stimulus labels and stimulus and button press times
    % must already be set.
    % See also rpe.fitExGauss rpe.exGaussPdf

    assert(~isempty(obj.stimulus_times), ...
        'RPE:TrainPdf:MissingStimulusTimes', ...
        'Cannot train response PDF with no stimuli. ');
    assert(~isempty(obj.stimulus_labels), ...
        'RPE:TrainPdf:MissingStimulusLabels', ...
        'Cannot train response PDF with no stimulus labels. ');
    assert(~isempty(obj.buttonpress_times), ...
        'RPE:TrainPdf:MissingButtonPress', ...
        'Cannot train response PDF with no button presses. ');

    %% build the RT list
    obj.buildRTLlist();

    %% from that vector, fit an exGaussian
    % ex gaussian is a distribution arising from a random
    % variable that is the sum of a normally distributed
    % random variable and an exponentially distributed random
    % variable. The exGaussian three parameters: tau is the
    % parameter of the exponential and mu & sigma are the
    % mean and standard deviation of the normal distribution.
    %
    % These values are fit using maximum likelihood
    % estimation.
    [obj.mu,obj.sigma,obj.tau] = rpe.fitExGauss(obj.rt_list);
    obj.pdf_fcn = @(rt) rpe.exGaussPdf(rt, ...
        obj.mu,obj.sigma,obj.tau);

    %% build a density approximation at requested resolution
    t = obj.time_resolution:obj.time_resolution:obj.pdf_support;
    obj.pdf_est = obj.pdf_fcn(t);
end
function [HR, FAR, HRCI, FARCI] = estimatePerformance(obj, alph)
    % Estimate HR and FAR

    % Build Beta
    obj.buildBeta;

    % Build Response Scores
    obj.buildResponseScores;

    % Solve for HR and FAR
    %o = obj.beta\[obj.response_scores]';
    if nargin == 2,

```

```

        o = regress(obj.response_scores', obj.beta);
    else
        if nargin < 2,
            alph = .05;
        end
        [o, ci] = regress(obj.response_scores', obj.beta, alph);
        HRCI = ci(1,:);
        FARCI = ci(2,:);
    end

    HR = o(1);
    FAR = o(2);

    % Correct HR/FAR
    if HR > 1,
        HR = 1;
    elseif HR < 0,
        HR = 0;
    end
    if FAR > 1,
        FAR = 1;
    elseif FAR < 0,
        FAR = 0;
    end
end
end
methods (Access=private)
function buildRTLList(obj)
    % To estimate the rt pdf, we need a collection of RTs.
    % Because we don't know what stimuli evoked which
    % responses, some heuristic is needed. The collection of
    % RTs is built using the window method. This means we go
    % over each target and look at a window of time after
    % that target. The first response that happens in that
    % window is assumed to be evoked by that target, so their
    % difference is added to a collection of RTs.
    %
    % This will, of course, be wrong sometimes, but it's the
    % best we can do (that I can think of).

    %% Initialize the parallel pool, if possible/needed
    try
        gcp();
    catch
        % no parallel toolbox
    end
    %% find stimuli labeled as targets
    tar_times = obj.stimulus_times(obj.stimulus_labels==true);
    %% build a vector of response times
    rts = zeros(size(tar_times));
    bpt = obj.buttonpress_times;
    rw = obj.response_window;
    % Note: creating local versions of these variables allows
    % faster parallel execution.

    parfor iTar = 1:numel(tar_times),
        tt = tar_times(iTar);
        resp_idx = find(bpt < tt+rw(2) & bpt > tt+rw(1)); %#ok
        if numel(resp_idx) == 0,
            %miss.
            continue
        elseif numel(resp_idx)>1,

```

```

        warning('RPE:BuildRTDist:MultiResponse',...
            ['The stimulus at time %f is followed by more ',...
            'than one responses (%d). Taking only the ',...
            'first.'],tt,numel(resp_idx));
        resp_idx = resp_idx(1);
    end
    rts(iTar) = bpt(resp_idx)-tt;
end
rts = rts(rts~=0);
obj.rt_list = rts;
end
function buildBeta(obj)
    %Assemble the regression coefficients
    try
        gcp();
    catch
        %parallel not available
    end
    l_beta = zeros(numel(obj.stimulus_times),2);

    mxrt = obj.pdf_support;
    ost = obj.stimulus_times;
    osl = obj.stimulus_labels;
    tr = obj.time_resolution;
    l_pdf_fcn = obj.pdf_fcn;
    l_pdf_support = obj.pdf_support;

    parfor iStim = 1:numel(ost),
        % get si, the stim of interest and sj, the list of stimuli
        % whos responses could be attributed to si
        si = ost(iStim);
        idx_neighbor = ost >= (si-mxrt) & ...
            ost <= (si+mxrt);
        neighbor_times = ost(idx_neighbor);
        neighbor_labels = osl(idx_neighbor); %#ok

        % compute expected attribution from each of sj, partitioned
        % as hit contributions and fa contributions.
        b1 = 0;
        b2 = 0;
        for jNeighbor = 1:numel(neighbor_times),
            sj = neighbor_times(jNeighbor);
            % figure out times of responses that could be generated
            % by sj and could contribute to attribution of si
            t_min = max(si, sj);
            t_max = min(si+mxrt, sj+mxrt);
            t = t_min:tr:t_max;

            % now compute attribution for each t
            a = att(t,si, l_pdf_fcn, ost, l_pdf_support); %#ok

            % compute the contribution to si of a response
            % by sj conditioned on a response by sj.
            e = sum(tr.*l_pdf_fcn(t-sj).*a);

            lbl = neighbor_labels(jNeighbor);
            if lbl,
                b1 = b1+e;
            else
                b2 = b2+e;
            end
        end
    end
end

```

```

        l_beta(iStim,:) = [b1 b2];
    end
    obj.beta = l_beta;
end

function buildResponseScores(obj)
    % Attribute each response to possible evoking stimuli.
    obj.response_scores = zeros(size(obj.stimulus_times));
    for iResp = 1:numel(obj.buttonpress_times),
        t = obj.buttonpress_times(iResp);
        candidate_idx = obj.stimulus_times < t & ...
            obj.stimulus_times > t - obj.pdf_support;

        if ~any(candidate_idx),
            % Rogue buttonpress.
            continue;
        end

        st = t - obj.stimulus_times(candidate_idx);
        lik = obj.pdf_fcn(st);
        scores = lik./sum(lik);
        obj.response_scores(candidate_idx) = scores + ...
            obj.response_scores(candidate_idx);
    end
end
end
end

```

#### A.4 fitExGauss.m

```

function [mu,s,tau] = fitExGauss(rts)
% Finds the parameters of an ex-gaussian function given an rt
% distribution using maximum likelihood estimation.
% [mu,s,tau] = fitExGauss(rts)
%
% Input rts is an array of response times. For good results, this array
% should have at least 30 entries. An error is thrown if it has less than
% 2.
%
% Outputs:
%
% mu and s the mean and standard deviation for the gaussian part of the
% exgaussian.
% tau is the parameter of the exponential part of the exgaussian.
%
% Written by Benjamin Files.
%
% References
% Palmer et al., (2011) What are the Shapes of Response Time
% Distributions in Visual Search?, Exp Psychol Hum Percept Perform.;
% 37(1): 58-71. doi:10.1037/a0020747
%
% Van Zandt, T. (2000). How to fit a response time distribution.
% Psychonomic Bulletin & Review, 7(3), 424-465.
%
% Inspired by DISTRIB toolbox of Yves Lacouture
% http://www.psy.ulaval.ca/?pid=1529
%

if numel(rts) < 3,
    error('Button:FitExGauss:NotEnoughSamples',...

```

```

        ['Cannot fit an ex-gaussian with less than 3 response times.
',...
        'only %d were provided.'],numel(rts));
end
if numel(rts) < 30,
    warning('Button:FitExGauss:FewSamples',...
        ['Fitting an ex-gaussian with %d samples. A fit will be ',...
        'provided, but the fit quality might be poor.'],numel(rts));
end

% These initial values are recommended in DISTRIB:
tauInit = std(rts)*.8;
muInit = mean(rts)-tauInit;
sigInit = sqrt(var(rts)-(tauInit.^2));

% Alternatively, use method of moments (e.g. Olivier, J., & Norberg, M.
% M. (2015). Positively Skewed Data: Revisiting the Box-Cox Power
% Transformation. International Journal of Psychological Research, 3(1),
% 68-77.
%
% This method is more accurate but also brittle (it has a lot of weird
% edges that result in nonsense estimates).
%
% g = skewness(rts);
% s = std(rts);
% m = mean(rts);
%
% muInit = m-s*(g/2)^(1/3);
% sigInit = sqrt( s^2*(1-(g/2)^(2/3)) );
% tauInit = s*(g/2)^(1/3);

start = [muInit,sigInit,tauInit];

% Setting bounds appropriately is tricky. In particular, if we let tau
% get too small, we overflow, because tau appears in the denominator of
% the eventual expression of the exGaussian PDF. max([0.01, min(rts)]) is
% mostly from trial-and-error.
%
% An alternative approach (not implemented) might be to check if MLE is
% trying to use a very small tau and instead of erroring, instead default
% to a normal distribution.
lb = [min(rts) min(rts) max([0.01, min(rts)])];
ub = [max(rts) lb(2)+range(rts) lb(end)+range(rts)];

too_low = start < lb;
start(too_low) = lb(too_low)+eps;

too_high = start > ub;
start(too_high) = ub(too_high)-eps;

ss = statset(@mlecustom);
ss.MaxFunEvals = 200*numel(rts);
ss.MaxIter = 200*numel(rts);
ss.FunValCheck = 'on';
phat = mle(rts,'pdf',@rpe.exGaussPdf,'start',start,'lowerbound',...
    lb,'upperbound',ub,'options',ss);
mu = phat(1);
s = phat(2);
tau = phat(3);
end

```

## A.5 exGaussPdf.m

```
function p = exGaussPdf(x,mu,s,tau)
% EXGAUSSPDF a probability density function for the exgaussian
% distribution.
% p = exGaussPdf(x,mu,s,tau)
%
% Mu, s and tau should be real numbers not less than zero.  Throws an
% error if not.
% Note, this blows up if tau is too small.
% x is the time(s) for which the probability density is requested.
% mu and s are the mean and standard deviation for the gaussian part of
% the exgaussian.
% tau is the parameter of the exponential part of the exgaussian.
% x, mu, s and tau are all assumed to have the same units.
%
% Written by Benjamin Files

% validate input.
ip = inputParser;
ip.addRequired('x',@isnumeric);
ip.addRequired('mu',@checkInput);
ip.addRequired('s',@checkInput);
ip.addRequired('tau',@checkInput);
ip.parse(x,mu,s,tau);
x = ip.Results.x;
mu = ip.Results.mu;
s = ip.Results.s;
tau = ip.Results.tau;

% check for overflow
tmp = mu/tau + s^2/(2*tau.^2) - x/tau;
maxtmp = log(realmax);
if any(tmp>=maxtmp),
    warning('RPE:ExGaussPdf:BigExpPart',...
        'A value exceeded max allowed.  Results will be approximate.');
```

$$\text{tmp}(\text{tmp} \geq \text{maxtmp}) = \text{maxtmp}/100;$$

```
end

% compute the pdf
pE = exp(tmp);
pG = normcdf( (x - mu - (s.^2/tau))/abs(s));
p = (1/tau).*pE.*pG;

if any (p<=eps),
    p(p<=eps) = eps;
end
end

function ok = checkInput(v)
ok = all([isnumeric(v),~isnan(v),~isinf(v),v>0]);
end
```

## List of Symbols, Abbreviations, and Acronyms

---

ANOVA	analysis of variance
EEG	electroencephalography
FAR	false-alarm rate
HR	hit rate
PDF	probability density function
RPE	RSVP performance estimator
RSVP	rapid serial visual presentation
RT-PDF	response-time probability density function

1 DEFENSE TECHNICAL  
(PDF) INFORMATION CTR  
DTIC OCA

2 DIR ARL  
(PDF) IMAL HRA  
RECORDS MGMT  
RDRL DCL  
TECH LIB

1 GOVT PRINTG OFC  
(PDF) A MALHOTRA

1 ARL  
(PDF) RDRL HRB B  
T DAVIS  
BLDG 5400 RM C242  
REDSTONE ARSENAL AL  
35898-7290

8 ARL  
(PDF) SFC PAUL RAY SMITH  
CENTER  
RDRL HRO COL H BUHL  
RDRL HRF J CHEN  
RDRL HRA I MARTINEZ  
RDRL HRR R SOTTILARE  
RDRL HRA C A RODRIGUEZ  
RDRL HRA B G GOODWIN  
RDRL HRA A C METEVIER  
RDRL HRA D B PETTIT  
12423 RESEARCH PARKWAY  
ORLANDO FL 32826

1 USA ARMY G1  
(PDF) DAPE HSI B KNAPP  
300 ARMY PENTAGON  
RM 2C489  
WASHINGTON DC 20310-0300

1 USAF 711 HPW  
(PDF) 711 HPW/RH K GEISS  
2698 G ST BLDG 190  
WRIGHT PATTERSON AFB OH  
45433-7604

1 USN ONR  
(PDF) ONR CODE 341 J TANGNEY  
875 N RANDOLPH STREET  
BLDG 87  
ARLINGTON VA 22203-1986

1 USA NSRDEC  
(PDF) RDNS D D TAMILIO  
10 GENERAL GREENE AVE  
NATICK MA 01760-2642

1 OSD OUSD ATL  
(PDF) HPT&B B PETRO  
4800 MARK CENTER DRIVE  
SUITE 17E08  
ALEXANDRIA VA 22350

ABERDEEN PROVING GROUND

13 DIR ARL  
(PDF) RDRL HR  
J LOCKETT  
P FRANASZCZUK  
K MCDOWELL  
K OIE  
RDRL HRB  
D HEADLEY  
RDRL HRB C  
J GRYNOVICKI  
RDRL HRB D  
C PAULILLO  
RDRL HRF A  
A DECOSTANZA  
J D CANADY  
RDRL HRF AA  
B T FILES  
RDRL HRF B  
A EVANS  
RDRL HRF C  
J GASTON  
RDRL HRF D  
A MARATHE